



Sentiment Analysis

A Probabilistic Approach

S. A. Gieske S. Laan D. S. Ten Velthuis
C. R. Verschoor A. J. Wiggers

Faculty of Science (FNWI)
University of Amsterdam

February 2, 2012



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



The goal of the project

Project Description

Performing sentiment analysis on messages about the EO

- Classification Sentiment vs. Non Sentiment
- Classification Positive vs. Negative



Outline

- 1 The goal
- 2 Approach**
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion

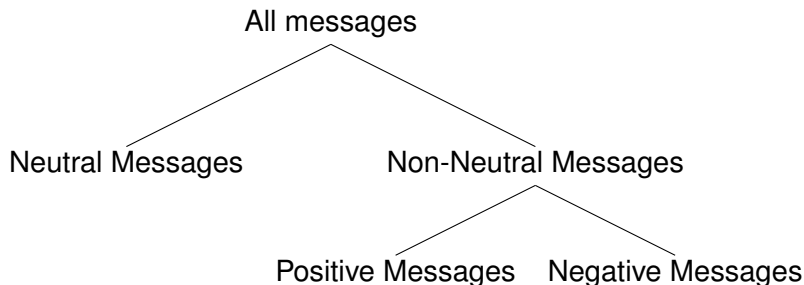


Approach

- Preprocessing of the data
- Perform machine learning algorithms on data
- Use best algorithms to classify real time on server



Hierarchical Classification





Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing**
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion



Outline

3 Data Preprocessing

- Dataset Analysis
- Data Cleaning
- Data Reduction



Dataset Analysis

Dataset messages EO

10.000 messages, 19 features per message

Only 3 features used:

- Source
- Sentiment
- Message contents



Data Cleaning

- Shorten words, ex. hahaha to haha
- Stemmer



Data Reduction

- Only use Twitter messages (83% of all messages)
- Remove articles, reference words and prepositions
- Substitute smileys with words
- Remove some punctuation marks (ex. not ! ?)



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification**
- 5 Webserver Framework
- 6 Conclusion



Outline

4 Classification

- Weighted Sum Probability
- Perceptron
- Support Vector Machine
- Naive Bayes
- Multiclassification with Perceptron
- Entropy
- Neural Network



Classification



Weighted Sum Probability

- Create tokens
- Assign sentiment probabilities to tokens

$$P(word) = \frac{\sum word \in C_1}{\sum word \in C_1 \cup C_2} \quad (1)$$

- Assign sentiment probabilities to sentences

$$P(s) = \frac{1}{n} \sum_{w \in s} P(w) \quad (2)$$



Perceptron

Algorithm

Train linear treshold

Input : Sentence probabilities, sentence values

Output : Treshold



Results & Conclusion

Results : High precision OR recall, never both

Conclusion : Lineaire threshold not good enough



Support Vector Machine

Algorithm

Fit in featurespace that binds features to classes

Input : Features in vector

Output : Number belonging to class



Results & Conclusion

Results : Not very good recall/precision/accuracy

Conclusion : Fit can not be made on these features
or more data needed to find clear boundary



Naive Bayes



Results & Conclusion

Results :

Conclusion :



Multiclassification with Perceptron

Multiclassification with Perceptron



Multiclassification with Perceptron

Results & Conclusion

Results :

Conclusion :



Entropy



Entropy



Entropy

Results & Conclusion

Results :

Conclusion :



Neural Network



Results & Conclusion

Results :

Conclusion :



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework**
- 6 Conclusion



Webserver Framework

Request (HTML) → Server (PHP/PYTHON) → Result (XML)

Request `http://url.com/?dataset=1&message=De EO is cool!`

Result XML File (Containing: Status, Message, Sentiment, Accuracy, Precision, Recall)



Demo

Action...



Outline

- 1 The goal
- 2 Approach
- 3 Data Preprocessing
- 4 Classification
- 5 Webserver Framework
- 6 Conclusion**



Conclusion



Conclusion